

プログラム解析による暗号通 信文の解析方法

ネットエージェント株式会社

杉浦 隆幸

概要

- 日ごろ使っているソフトウェアがどのような通信をしているか知っていますか。実際は送りたいくない情報を送っているかもしれません、送りたいくないような情報を送っているようなソフトウェアほど、通信を暗号化し、技術的に何を送っているのかを解明することが難しくなっています。難しい部分もありますが、これらを判明させていく手法を順を追って紹介します。

- 独自プロトコルソフトは何を送っているのでしょうか？
 - メッセンジャー
 - ウイルス対策ソフトのデータ
 - ブラウザ以外のSSL通信
 - Winny、Share、SpkypeなどのP2Pソフトウェア
- クイズがあるので分かったら教えてください。

手法

- 手法1: パケットをキャプチャする
- 手法2: プロキシサーバ、中継サーバで記録する
- 手法3: プログラムのメモリーイメージを見てみる
- 手法4: プログラムにディバッガを当てて暗号通信の使用部分に来たときのデータを見てみる。

対象

暗号通信をして情報を隠したいようなソフトって
どんなのことがあるソフト？

- スパイウェア
 - 変な動作をするソフトウェア
- ボット
 - 暗号化してボットネットにつながっている場合
- P2Pソフト
 - WinnyとかShareとかPerfect Darkとか

できる限り楽をして・・・

- 解析には段階がある。

できれば楽な方法でやりたい。

- 1. 平分通信ならパケットキャプチャで
- 2. 実行ファイルの文字列にあるならstringsで
- 3. プログラムのメモリーイメージでよいならそれで
- 4. ソースがあるのならそれを読んで
- 5. 仕方ないので、バイナリのプログラムを解析して

パケットキャプチャの前に NetAgent

ネットエージェント株式会社

- TCPviewで調べるポート確認
- <http://www.microsoft.com/technet/sysinternals/utilities/TcpView.msp>
 - 使っているプログラムがグラフィカルに出るnetstat

TCPView - Sysinternals: www.sysinternals.com

File Options Process View Help

Process	Protocol	Local Address	Remote Address	State
[System Process]:0	TCP	aka.m.nd.to:2516	www.ushio.co.jp:http	TIME_WAIT
[System Process]:0	TCP	aka.m.nd.to:2458	a61-208-229-161.deploy.akamaitechnologies.com...	TIME_WAIT
firefox.exe:2860	TCP	aka:1034	localhost:1035	ESTABLISHED
firefox.exe:2860	TCP	aka:1035	localhost:1034	ESTABLISHED
firefox.exe:2860	TCP	aka:1036	localhost:1037	ESTABLISHED
firefox.exe:2860	TCP	aka:1037	localhost:1036	ESTABLISHED
LSASS.EXE:260	UDP	aka:3432	.*.*	.
msnmsgr.exe:1316	UDP	aka:1036	.*.*	.
mstask.exe:916	TCP	aka:1029	aka:0	LISTENING
PCastMediaServe:1340	TCP	aka:7999	aka:0	LISTENING
PCastMediaServe:1340	TCP	aka:8000	aka:0	LISTENING
PCastMediaServe:1340	TCP	aka:9001	aka:0	LISTENING
PCastMediaServe:1340	TCP	aka:1040	localhost:9001	ESTABLISHED
PCastMediaServe:1340	TCP	aka:9001	localhost:1040	ESTABLISHED
PCastMediaServe:1340	UDP	aka:5353	.*.*	.
sqlservr.exe:720	TCP	aka.ms-sql-s	aka:0	LISTENING
sqlservr.exe:720	UDP	aka.ms-sql-m	.*.*	.
svchost.exe:440	TCP	aka:epmap	aka:0	LISTENING
System:8	TCP	aka:microsoft-ds	aka:0	LISTENING
System:8	TCP	aka.m.nd.to:netbios-ssn	aka:0	LISTENING
System:8	TCP	aka.m.nd.to:netbios-ssn	aka:0	LISTENING
System:8	TCP	aka.m.nd.to:netbios-ssn	aka:0	LISTENING
System:8	UDP	aka:microsoft-ds	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-ns	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-dgm	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-ns	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-dgm	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-ns	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-dgm	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-ns	.*.*	.
System:8	UDP	aka.m.nd.to:netbios-dgm	.*.*	.
vpnclient.exe:992	TCP	aka:9999	aka:0	LISTENING
WINLOGON.EXE:220	UDP	aka:3942	.*.*	.

パケットキャプチャ

パケット解析 ヘルプ
Copyright(c) NetAgent Co.,Ltd. 2006

メール Web Web画像 メッセージャー Windowsファイル共有 SQL_server oracle VoIP Notes
TCP通信 通信内容記録 ユーザ・サーバ別 詳細全文検索 インデックス検索 grep IDS One Point Wall ウィルス

/O/g1169636403/pkt/2007-01-24_21_27[22].pkt ▲ ファイルの変更

パケットを解析し表示 ▼

フィルタ: (ip.addr eq 10.0.0.153 and ip.addr eq 221) and (tcp.port eq 3602 and tcp.port eq 42)

フレーム データリンク ネットワーク トランスポート アプリケーション 全表示
 詳細 詳細 詳細 詳細

表示件数 500 ▼ 件 解析 ヘルプ

```
Data (11 bytes)
|
0000  ff db  de 9a d7  ...

Data (74 bytes)
0000  64 a3 38 41 67 94 9a 6b a1 91 2b 2e e0 10 22 77  d.8Ag..k..+...~w
0010  96 ca e0 48 71 d6 a6 35 d1 a9 39 ce 91 8a a9 7f  ...Hq..5..9.....
0020  2b a2 86 c5 86 e2 16 f1 9f d1 1e c7 6b a0 94 bd  +.....k...
0030  11 13 63 6d 33 1e 42 dd 78 4c 0e bc bc 17 f9 93  ..cm3.B.xL.....
0040  c1 e3 7a 80 61 93 c6 42 53 ab  ..z.a..BS.

Data (36 bytes)
0000  51 43 66 03 4e 7b b3 b2 c7 64 ea 1e e9 e8 0c 85  QCf.N{...d.....
0010  4e f3 21 54 7e 75 a3 e1 e4 47 ca ee 3f b8 c9 69  N.!T~u...G..?..i
0020  a8 b0 67 4f  ..g0

=== END ===
```

クイズ: この通信は何でしょう?

パケットキャプチャ

•[http://www.google-analytics.com/__utm.gif?utmwv=1&utmz=1208086912&utmcs=iso-2022-jp&utmsr=1280x1024&utmssc=32-bit&utmul=ja&utmje=1&utmfl=8.0&utmcn=1&utmdt=Sun%20Microsystems%20-%20%E3%82%A2%E3%83%97%E3%83%AA%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%81%AE%E6%8B%A1%E5%BC%B5&utmhn=jp.sun.com&utmr=http://www.google.co.jp/search?hl=ja&q=off_t&btnG=Google+%E6%A4%9C%E7%B4%A2&lr=lang_ja&utmp=/products/software/solaris/wp/Sol_file/sol_3.html&utmac=UA-150017-1&utmcc=__utma%3D164123876.1208086912.1169641053.1169641053.1169641053.1%3B%2B__utmb%3D164123876%3B%2B__utmc%3D164123876%3B%2B__utmz%3D164123876.1169641053.1.1.utmccn%3D\(organic\)%7Cutmcsr%3Dgoogle%7Cutmctr%3Doff_t%7Cutmcmd%3Dorganic%3B%2B](http://www.google-analytics.com/__utm.gif?utmwv=1&utmz=1208086912.1169641053.1169641053.1169641053.1%3B%2B__utmb%3D164123876%3B%2B__utmc%3D164123876%3B%2B__utmz%3D164123876.1169641053.1.1.utmccn%3D(organic)%7Cutmcsr%3Dgoogle%7Cutmctr%3Doff_t%7Cutmcmd%3Dorganic%3B%2B)

妙に長いアクセス

結果



パケットキャプチャ

Googleのトラッキング用のURL
で透過1x1ピクセルの画像表示

POST

```
utmwv=1
utmn=1208086912
utmcs=iso-2022-jp
utmsr=1280x1024
utmssc=32-bit
utmul=ja
utmje=1
utmfl=8.0
utmcn=1
utmdt=Sun Microsystems - アプリケーションの拡張
utmhn=jp.sun.com
utmr=http://www.google.co.jp/search?hl=ja
q=off_t
btnG=Google 検索
lr=lang_ja
utmp=/products/software/solaris/wp/Sol_file/sol_3.html|
utmact=UA-150017-1
utmcc=__utma=164123876.1208086912.1169641053.1169641053.116964105
```

プログラムのメモリーイメージを解析

- 実際にやってみましょう。
- ProcessWalkerか、Ollydbgを使います

今日来た人が見られる特典です。

プログラムのメモリエイジーを解析

The screenshot displays the ProcessWalker application interface. On the left, a tree view shows the process 'eikoNavi.exe' selected, with its memory structure expanded to show '仮想メモリ' (Virtual Memory) and 'ヒープウォーク' (Heap Walk). The main window shows a list of processes with columns for Name, PID, User, Working Set, Private Memory, CPU Time, Start Time, Thread Count, Handle Count, and Path. A 'メモリダンプ' (Memory Dump) window is open, showing a hex dump of memory starting at offset 00FA0000. The dump includes hexadecimal values and their corresponding ASCII characters, such as '0123456789ABCDEF' and various control characters and symbols.

名前	PID	ユーザー	ワーキングセット	プライベートメモリ	CPU時間	開始時間	スレッド数	ハンドル数	パス
ttermpro.exe	4016	MWlumin	1,167,360	6,492,160	0:00:04.734	2007/1/25 11:18:14.453	2	78	C:\Program File...
firefox.exe	4316	MWlumin	90,550,272	78,426,112	0:12:06.687	2007/1/31 19:13:12.421	23	516	C:\Program File...
vncviewer.exe	4420	MWlumin	6,324,224	5,582,848	0:00:09.984	2007/1/22 13:41:47.812	3	81	C:\Program File...
ttermpro.exe	4524	MWlumin							
Hidemaru.exe	4536	MWlumin							

ADDRESS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	0123456789ABCDEF	
00FE4650	A4	43	FB	00	00	00	00	00	00	00	00	00	00	CC	34	FE	00	.C.....4..
00FE4680	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00@F..4..
00FE4670	A8	46	FE	00	00	00	00	00	00	00	00	00	00	CC	34	FE	00	.F.....@F..4..
00FE4680	88	08	4F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..0..e.....
00FE4690	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00p.E.....
00FE46A0	01	00	00	00	16	00	00	00	70	E9	45	00	00	00	00	00	00h.B.
00FE46B0	6C	45	46	00	F8	45	FE	00	2A	00	00	00	00	88	81	42	00	IEF..E.*..h.B.
00FE46C0	00	00	00	00	F8	AD	42	00	8C	5E	FE	00	5C	12	FA	00	00B.[..¥..
00FE46D0	C4	5B	FE	00	08	00	00	80	60	00	00	00	00	00	00	00	00
00FE46E0	42	00	00	00	01	00	00	00	33	00	00	00	00	88	74	74	70	B.....3...http
00FE46F0	3A	2F	2F	70	83	38	2E	32	63	68	2E	6E	65	74	2F	74		://pc8.2ch.net/t
00FE4700	65	73	74	2F	72	65	61	64	2E	63	67	69	2F	73	65	63		est/read.cgi/sec
00FE4710	2F	31	30	39	36	35	35	37	38	39	36	2F	6C	35	30	00		/1096557896/150.
00FE4720	16	00	00	00	7C	4F	4E	00	84	E6	06	01	80	97	00	01	[ON].....
00FE4730	00	00	00	00	C8	8D	89	05	D8	7D	00	01	10	00	00	00	}
00FE4740	10	00	00	00	27	00	00	00	01	00	00	00	14	00	00	00	
00FE4750	52	65	73	52	61	6E	67	65	4E	65	77	50	4D	65	6E	75		ResRangeNewPMMenu
00FE4760	48	74	65	6D	00	00	00	00	82	00	00	00	E8	07	46	00		Item.....F.
00FE4770	0C	A7	FB	00	50	47	FE	00	00	00	00	00	00	00	00	00	PG.....
00FE4780	5C	98	FE	00	00	00	00	00	00	00	00	00	00	00	00	00	
00FE4790	01	00	00	00	08	87	41	00	18	87	41	00	08	F1	FA	00	A..A.....
00FE47A0	00	00	00	00	01	00	02	02	00	01	02	2D	00	00	00	00	A..A.....
00FE47B0	00	98	FE	00	00	00	00	00	00	00	00	00	1A	01	00	00	IA 01 00 00
00FE47C0	00	00	00	00	D4	F3	FA	00	00	00	00	00	00	00	00	00	
00FE47D0	CC	34	FE	00	00	00	00	00	00	00	00	00	00	00	00	00	4.....
00FE47E0	00	00	00	00	1C	48	FE	00	00	00	00	00	00	00	00	00	H.....@F.
00FE47F0	CC	34	FE	00	00	00	00	00	00	00	00	00	00	00	00	00	4.....0..e.....
00FE4800	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00FE4810	00	00	00	00	01	00	00	00	16	00	00	00	70	E9	45	00	p.E.....
00FE4820	00	00	00	00	6C	45	46	00	6C	47	FE	00	2A	00	00	00	IEF..G.*..
00FE4830	68	81	42	00	00	00	00	18	18	45	00	64	5E	FE	00	00		h..B.....E.d~
00FE4840	5C	4F	FE	00	00	00	00	08	00	80	60	00	00	00	00	00		¥0.....
00FE4850	00	00	00	00	6A	00	00	E0	C4	44	00	00	00	00	00	00	j.....D.....
00FE4860	00	00	00	00	18	2D	5A	05	C0	48	FE	00	78	99	00	01	H..x.....
00FE4870	00	00	00	00	00	00	00	00	20	00	CC	00	00	00	00	00	
00FE4880	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

プログラム解析ツール

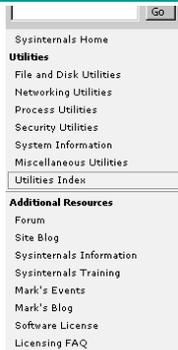
- ollydbg (free)
 - <http://www.ollydbg.de/>
- IDA Pro (470 USD or 895 USD)
 - <http://www.datarescue.com/>
 - FAXを送ってクレジットカードで買う
- プログラム言語 (cとかperlとか)

プログラム解析を始めよう！

- まずは楽にできるところから
 - Regmon, filemon

レジストリの読み書きを確認

- Regmon
- <http://www.microsoft.com/technet/sysinternals/utilities/Regmon.mspix>



RegMon for Windows v7.04

By Mark Russinovich and Bryce Cogswell

Published: November 1, 2006

Introduction

Note: Filemon and Regmon have been replaced by Process Monitor on versions of Windows starting with Windows 2000 SP4, Windows XP SP2, Windows Server 2003 SP1, and Windows Vista. Filemon and Regmon remain for legacy operating system support, including Windows 9x.

Regmon is a Registry monitoring utility that will show you which applications are accessing your Registry, which keys they are accessing, and the Registry data that they are reading and writing - all in real-time. This advanced utility takes you one step beyond what static Registry tools can do, to let you see and understand exactly how programs use the Registry. With static tools you might be able to see what Registry values and keys changed. With Regmon you'll see how the values and keys changed.

Regmon works on Windows NT/2000/XP/2003, Windows 95/98/Me and Windows 64-bit for x64.

ID	Time	Process	Request	Path	Result	Other
11453	7.2750939	lsass.exe	OpenKey	HKLM\SECURITY\Policy\SecDesc	SUCCESS	Key 0x1239408
11454	7.2751105	lsass.exe	OpenKey	HKLM\SECURITY\Policy\SecDesc	SUCCESS	NONE
11455	7.27513675	lsass.exe	CloseKey	HKLM\SECURITY\Policy\SecDesc	SUCCESS	Key 0x1239408
11456	7.2751999	lsass.exe	OpenKey	HKLM\SECURITY\Policy	SUCCESS	Key 0x1148788
11457	7.50898723	inohost.exe	OpenKey	HKLM\System\CurrentControlSet\Services\Debug	SUCCESS	Key 0x11509F88
11458	7.50898722	inohost.exe	QueryValue	HKLM\System\CurrentControlSet\Services\Debug\hkey	NOTFOUND	
11459	7.50898722	inohost.exe	QueryValue	HKLM\System\CurrentControlSet\Services\Debug\hkey	NOTFOUND	
11460	7.50898723	inohost.exe	CloseKey	HKLM\System\CurrentControlSet\Services\Debug	SUCCESS	Key 0x11509F88
11461	7.50877489	inohost.exe	OpenKey	HKLM\SYSTEM\ControlSet002\Services\idont	SUCCESS	Key 0x11509F88
11462	7.50877493	inohost.exe	OpenKey	HKLM\SYSTEM\ControlSet002\Services\idont\DebugLevel	NOTFOUND	
11463	7.50877489	inohost.exe	CloseKey	HKLM\SYSTEM\ControlSet002\Services\idont	SUCCESS	Key 0x11509F88
11464	8.58465268	inohost.exe	OpenKey	HKLM\System\CurrentControlSet\Services\Debug	NOTFOUND	Key 0x11509F88
11465	8.58465268	inohost.exe	QueryValue	HKLM\System\CurrentControlSet\Services\Debug\hkey	NOTFOUND	
11466	8.58465268	inohost.exe	QueryValue	HKLM\System\CurrentControlSet\Services\Debug\hkey	NOTFOUND	
11467	8.58419832	inohost.exe	CloseKey	HKLM\System\CurrentControlSet\Services\Debug	SUCCESS	Key 0x11509F88
11468	8.58419831	inohost.exe	OpenKey	HKLM\SYSTEM\ControlSet002\Services\idont	SUCCESS	Key 0x11509F88
11469	8.58419831	inohost.exe	QueryValue	HKLM\SYSTEM\ControlSet002\Services\idont\DebugLevel	NOTFOUND	
11470	8.58423237	inohost.exe	CloseKey	HKLM\SYSTEM\ControlSet002\Services\idont	SUCCESS	Key 0x11509F88
11471	8.60869843	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	BUFDIVLW	
11472	8.60869872	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	BUFDIVLW	
11473	8.60869899	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	BUFDIVLW	
11474	8.60799802	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	BUFDIVLW	
11475	8.60869843	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	BUFDIVLW	
11476	8.60800747	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\ControlSet002\Services\Tcpip\Linkage\Bind	SUCCESS	"Device\BACD088...
11477	8.61002626	OUTLOOK.EXE	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\...	SUCCESS	Key 0x11509F88
11478	8.61002626	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\...	SUCCESS	0x1
11479	8.61002626	OUTLOOK.EXE	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\...	SUCCESS	0x20A88D...



regmon

#	Time	Process	Request	Path	Result	0
6755	39.65703201	epmworker.exe:1472	CloseKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	
6756	40.05735397	epmworker.exe:1472	OpenKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	Ac
6757	40.05737686	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6758	40.05739594	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6759	40.05742645	epmworker.exe:1472	CloseKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	
6760	40.44413757	epmworker.exe:1472	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Setup	SUCCESS	Ac
6761	40.44416046	epmworker.exe:1472	QueryValue	HKLM\Software\Microsoft\Windows\CurrentVersion\Setup\LogLevel	SUCCESS	0x
6762	40.44417572	epmworker.exe:1472	QueryValue	HKLM\Software\Microsoft\Windows\CurrentVersion\Setup\LogPath	NOT FOUND	
6763	40.44419479	epmworker.exe:1472	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Setup\AppLogLevels	NOT FOUND	
6764	40.44422531	epmworker.exe:1472	CloseKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Setup	SUCCESS	
6765	40.44446182	epmworker.exe:1472	CreateKey	HKLM\System\CurrentControlSet\Control\DeviceClasses	SUCCESS	Ac
6766	40.44470978	SERVICES.EXE:248	CreateKey	HKLM\System\CurrentControlSet\Control\DeviceClasses	SUCCESS	Ac
6767	40.44473267	SERVICES.EXE:248	OpenKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{c332cccc-a02f-421d-a2ac...	NOT FOUND	
6768	40.44475174	SERVICES.EXE:248	CloseKey	HKLM\System\CurrentControlSet\Control\DeviceClasses	SUCCESS	
6769	40.44482040	epmworker.exe:1472	OpenKey	HKLM\System\CurrentControlSet\Control\DeviceClasses\{C332CCCC-A02F-421D-A2...	NOT FOUND	
6770	40.44485092	epmworker.exe:1472	CloseKey	HKLM\System\CurrentControlSet\Control\DeviceClasses	SUCCESS	
6771	40.45773315	epmworker.exe:1472	OpenKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	Ac
6772	40.45775604	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6773	40.45777512	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6774	40.45780182	epmworker.exe:1472	CloseKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	
6775	40.68435287	sqlmangr.exe:1376	OpenKey	HKLM\System\CurrentControlSet\Control\ComputerName	SUCCESS	Ac
6776	40.68438721	sqlmangr.exe:1376	OpenKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	SUCCESS	Ac
6777	40.68440628	sqlmangr.exe:1376	QueryValue	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName\Co...	SUCCESS	"A
6778	40.68443298	sqlmangr.exe:1376	CloseKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	SUCCESS	
6779	40.68445587	sqlmangr.exe:1376	CloseKey	HKLM\System\CurrentControlSet\Control\ComputerName	SUCCESS	
6780	40.85813522	epmworker.exe:1472	OpenKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	Ac
6781	40.85815811	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6782	40.85821152	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6783	40.85824203	epmworker.exe:1472	CloseKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	
6784	41.25852585	epmworker.exe:1472	OpenKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	Ac
6785	41.25854874	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6786	41.25856781	epmworker.exe:1472	QueryValue	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI\WizC...	SUCCESS	0x
6787	41.25859833	epmworker.exe:1472	CloseKey	HKLM\Software\Toshiba\Mobile Connectivity Suite\Mobile Phone Monitor\CAPI	SUCCESS	

ファイルの読み書きを確認

- Filemon
- <http://www.microsoft.com/technet/sysinternals/SystemInformation/Filemon.mspix>

filemon

The screenshot shows the File Monitor application window with the following data:

#	Time	Process	Request	Path
867	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
868	2:29:54	internat.exe:1208	OPEN	C:\WINNT\system32\imejp.ime
869	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
870	2:29:54	internat.exe:1208	SET INFORMATION	C:\WINNT\system32\imejp.ime
871	2:29:54	internat.exe:1208	READ	C:\WINNT\system32\imejp.ime
872	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
873	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
874	2:29:54	internat.exe:1208	CLOSE	C:\WINNT\system32\imejp.ime
875	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
876	2:29:54	internat.exe:1208	OPEN	C:\WINNT\system32\imejp.ime
877	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
878	2:29:54	internat.exe:1208	SET INFORMATION	C:\WINNT\system32\imejp.ime
879	2:29:54	internat.exe:1208	READ	C:\WINNT\system32\imejp.ime
880	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
881	2:29:54	internat.exe:1208	QUERY INFORMATION	C:\WINNT\system32\imejp.ime
882	2:29:54	internat.exe:1208	CLOSE	C:\WINNT\system32\imejp.ime
883	2:29:54	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
884	2:29:54	explorer.exe:1172	OPEN	C:\WINNT\system32\taskmgr.exe
885	2:29:54	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
886	2:29:54	explorer.exe:1172	CLOSE	C:\WINNT\system32\taskmgr.exe
887	2:29:55	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
888	2:29:55	explorer.exe:1172	OPEN	C:\WINNT\system32\taskmgr.exe
889	2:29:55	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
890	2:29:55	explorer.exe:1172	CLOSE	C:\WINNT\system32\taskmgr.exe
891	2:29:56	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
892	2:29:56	explorer.exe:1172	OPEN	C:\WINNT\system32\taskmgr.exe
893	2:29:56	explorer.exe:1172	QUERY INFORMATION	C:\WINNT\system32\taskmgr.exe
894	2:29:56	explorer.exe:1172	CLOSE	C:\WINNT\system32\taskmgr.exe

プロセスの状態を取得

- Process walker
 - DLLやメモリーイメージが見られるので、
 - 通信内容が平文で存在したらすぐにわかる。

うさみみハリケーン

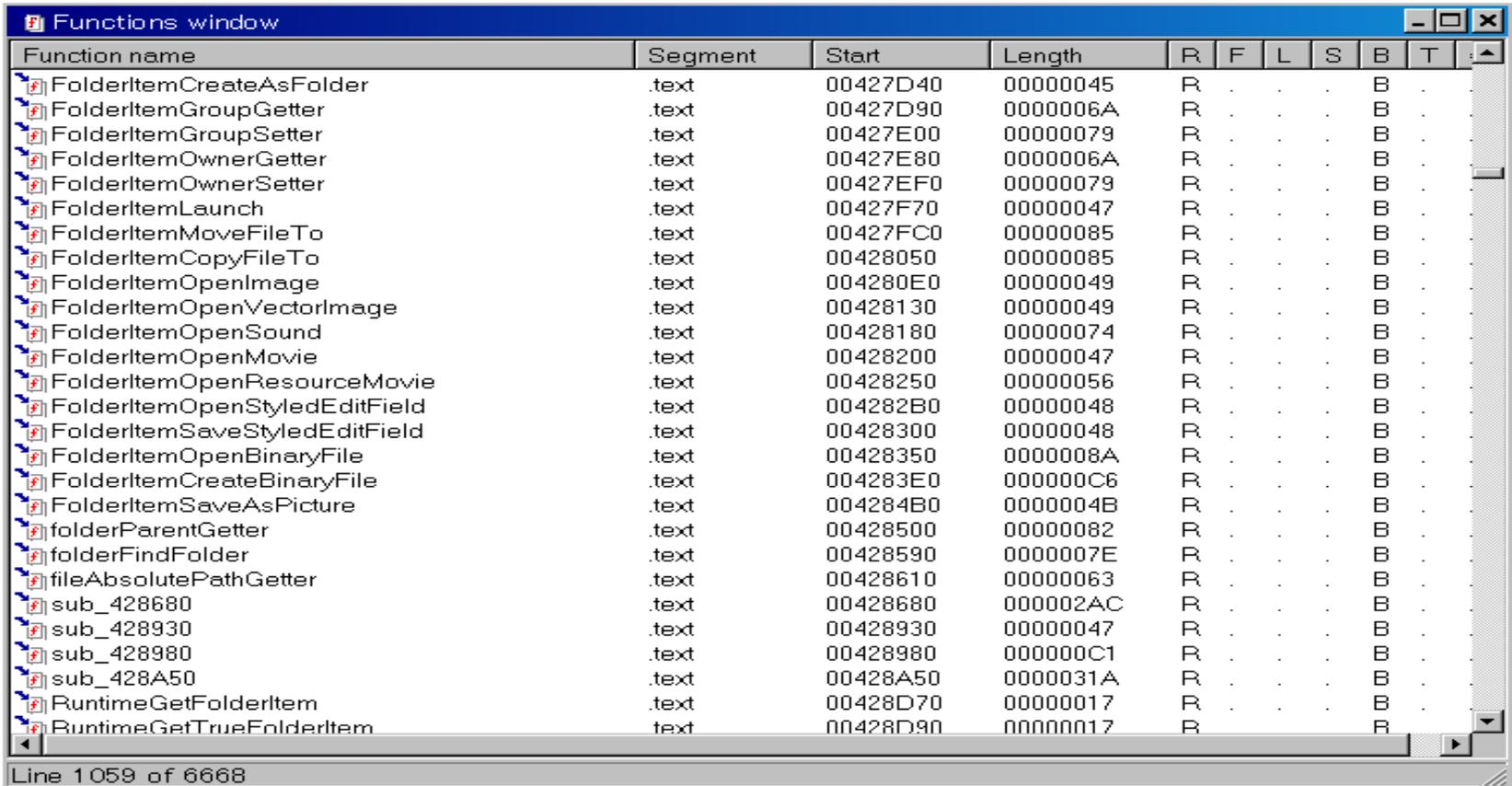
- PEヘッダの解析
- リソース解析
- DLLからのインポート関数一覧表示

スタティック解析 (IDA Pro)

- 実行ファイル、DLL、ocx等を読み込むと、1000～10000ぐらいのサブルーチンに分けてくれるので、それを一つ一つ読んでいく。
- IDA Proを使ってみよう

実際にデモしてみます

書かれた言語を知る



Function name	Segment	Start	Length	R	F	L	S	B	T
FolderItemCreateAsFolder	.text	00427D40	00000045	R	.	.	.	B	.
FolderItemGroupGetter	.text	00427D90	0000006A	R	.	.	.	B	.
FolderItemGroupSetter	.text	00427E00	00000079	R	.	.	.	B	.
FolderItemOwnerGetter	.text	00427E80	0000006A	R	.	.	.	B	.
FolderItemOwnerSetter	.text	00427EF0	00000079	R	.	.	.	B	.
FolderItemLaunch	.text	00427F70	00000047	R	.	.	.	B	.
FolderItemMoveFileTo	.text	00427FC0	00000085	R	.	.	.	B	.
FolderItemCopyFileTo	.text	00428050	00000085	R	.	.	.	B	.
FolderItemOpenImage	.text	004280E0	00000049	R	.	.	.	B	.
FolderItemOpenVectorImage	.text	00428130	00000049	R	.	.	.	B	.
FolderItemOpenSound	.text	00428180	00000074	R	.	.	.	B	.
FolderItemOpenMovie	.text	00428200	00000047	R	.	.	.	B	.
FolderItemOpenResourceMovie	.text	00428250	00000056	R	.	.	.	B	.
FolderItemOpenStyledEditField	.text	004282B0	00000048	R	.	.	.	B	.
FolderItemSaveStyledEditField	.text	00428300	00000048	R	.	.	.	B	.
FolderItemOpenBinaryFile	.text	00428350	0000008A	R	.	.	.	B	.
FolderItemCreateBinaryFile	.text	004283E0	000000C6	R	.	.	.	B	.
FolderItemSaveAsPicture	.text	004284B0	0000004B	R	.	.	.	B	.
folderParentGetter	.text	00428500	00000082	R	.	.	.	B	.
folderFindFolder	.text	00428590	0000007E	R	.	.	.	B	.
fileAbsolutePathGetter	.text	00428610	00000063	R	.	.	.	B	.
sub_428680	.text	00428680	000002AC	R	.	.	.	B	.
sub_428930	.text	00428930	00000047	R	.	.	.	B	.
sub_428980	.text	00428980	000000C1	R	.	.	.	B	.
sub_428A50	.text	00428A50	0000031A	R	.	.	.	B	.
RuntimeGetFolderItem	.text	00428D70	00000017	R	.	.	.	B	.
RuntimeGetTrueFolderItem	.text	00428D90	00000017	R	.	.	.	B	R

Line 1059 of 6668

クイズ：このプログラムの書かれた言語を当ててみましょう。

Javaバイナリ

The screenshot shows the IDA Pro interface. The main window displays assembly code for a function. The code includes variable declarations for registers and pointers, followed by a series of push instructions and a call to `ds:RegOpenKeyExA`. The strings window on the right lists various error messages and system paths, with `CLASSPATH` highlighted.

```
arg_4= dword ptr 8
arg_C= dword ptr 10h
hKey= dword ptr 1Ch
phkResult= dword ptr 20h
Data= byte ptr 24h
Buffer= byte ptr 128h
arg_8F4= dword ptr 8F8h
arg_10C4= dword ptr 10C8h
arg_1894= dword ptr 1898h
arg_2064= dword ptr 2068h
lpData= dword ptr 283Ch
arg_283C= dword ptr 2840h

mov     eax, 2838h
call    sub_4052E0
lea     ecx, [esp+hKey]
push    ecx                ; phkResult
push    20019h             ; samDesired
push    0                  ; ulOptions
push    offset SubKey     ; "Software¥¥JavaSoft¥¥Java Runtime Environm"...
push    80000002h        ; hKey
call    ds:RegOpenKeyExA
test   eax, eax
jz     short loc_401538
```

Address	Length	Ty...	String
...data:0040E008	00000013	C	Error loading: %s\n
...data:0040E01C	00000011	C	JNI_CreateJavaVM
...data:0040E030	0000001D	C	JNI_GetDefaultJavaVMInitArgs
...data:0040E050	00000022	C	Can't find JNI interfaces in: %s\n
...data:0040E074	0000000F	C	Executor Error
...data:0040E084	00000009	C	Executor
...data:0040E09C	0000002B	C	Software\JavaSoft\Java Runtime Environment
...data:0040E0C8	0000000F	C	CurrentVersion
...data:0040E0EC	00000027	C	Unrecognized minimum major version: %s
...data:0040E120	00000027	C	Unrecognized maximum major version: %s
...data:0040E148	0000000B	C	RuntimeLib
...data:0040E154	00000031	C	Failed reading value of registry key: RuntimeLib
...data:0040E188	0000002E	C	Could not find the Java Runtime Environment\n
...data:0040E1B8	0000004A	C	No recognized versions in: 'Software\JavaSoft\Java Runtime Environment\...
...data:0040E210	00000015	C	_JAVA_LAUNCHER_DEBUG
...data:0040E228	0000001E	C	__JAVA_LAUNCHER_DEBUG__\n
...data:0040E248	0000000A	C	CLASSPATH
...data:0040E254	00000005	C	.exe

VC++

The screenshot shows a debugger window with two main panes. The left pane displays assembly code for a function, and the right pane shows a 'Functions window' listing various subroutines.

Assembly Code (Left Pane):

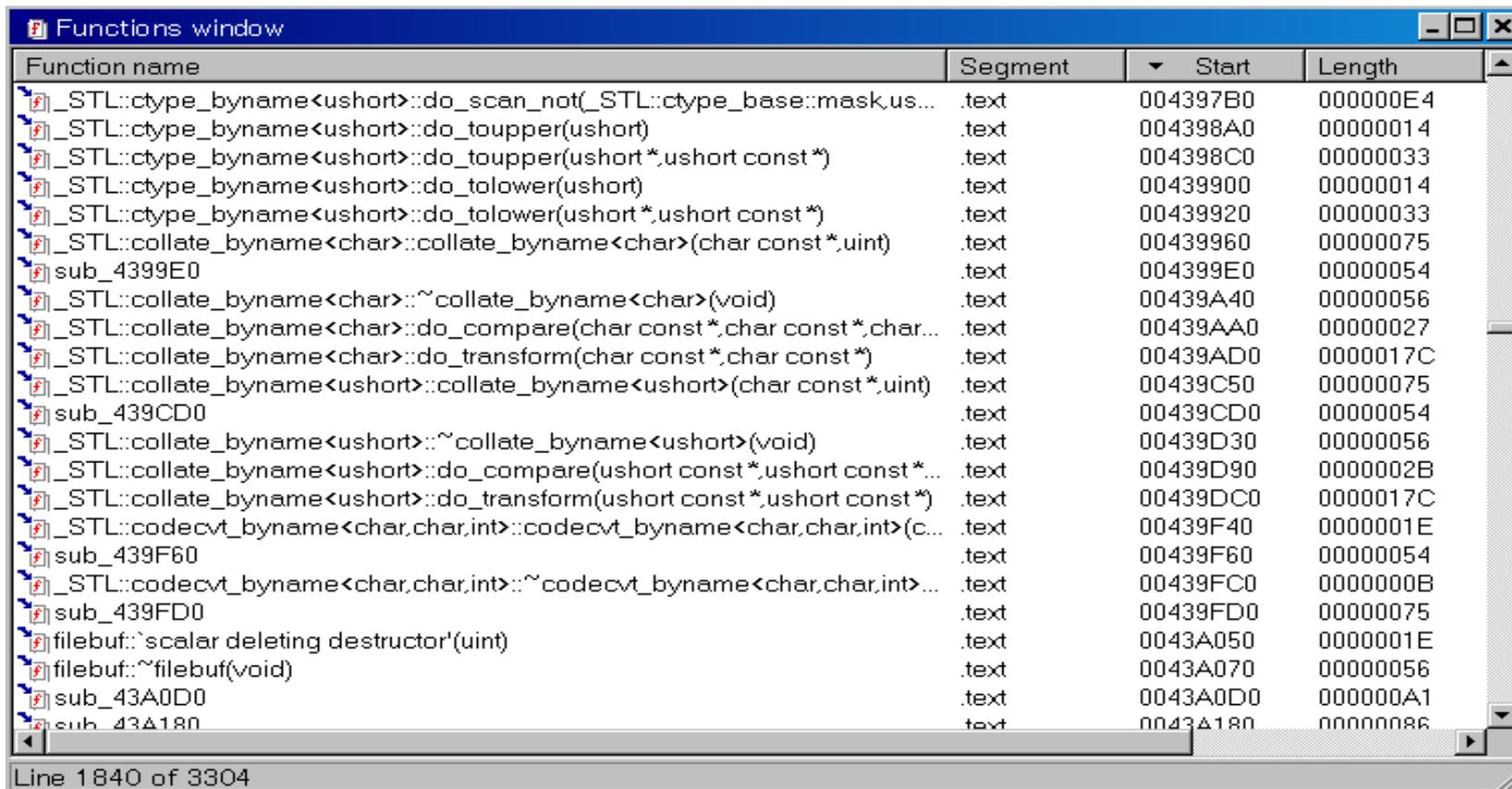
```
; Attributes: library function  
  
; int __stdcall WinMain(HINSTANCE hInst, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd) proc near  
_WinMain@16 proc near  
  
hInstance= dword ptr 4  
hPrevInstance= dword ptr 8  
lpCmdLine= dword ptr 0Ch  
nShowCmd= dword ptr 10h  
  
push [esp+nShowCmd]  
push [esp+4+lpCmdLine]  
push [esp+8+hPrevInstance]  
push [esp+0Ch+hInstance]  
call ?AfxWinMain@YGHPAUHINSTANCE_@  
retn 10h  
_WinMain@16 endp
```

Functions Window (Right Pane):

Function name	Segment	Start	Length	R	F	L	S	B	T
sub_4A2D9F	.text	004A2D9F	00000048	R
sub_4A3037	.text	004A3037	00000098	R
sub_4A30CF	.text	004A30CF	00000043	R
sub_4A3112	.text	004A3112	0000003F	R
ClmgeList::CreateObject(void)	.text	004A3151	00000032	R	.	L	.	.	.
sub_4A3183	.text	004A3183	00000006	R
DNameNode::DNameNode(void)	.text	004A3189	0000000D	R	.	L	.	.	.
sub_4A3196	.text	004A3196	0000001C	R
sub_4A31B2	.text	004A31B2	00000007	R
sub_4A31B9	.text	004A31B9	00000007	R
sub_4A31C0	.text	004A31C0	00000018	R
AfxThrowOleException(long)	.text	004A31D8	00000032	R	.	L	.	.	.
sub_4A320A	.text	004A320A	00000006	R	.	L	.	.	.
sub_4A3210	.text	004A3210	00000011	R
sub_4A3221	.text	004A3221	0000001C	R
sub_4A323D	.text	004A323D	0000002B	R	T
sub_4A3268	.text	004A3268	00000035	R	T
sub_4A329D	.text	004A329D	00000021	R
AfxThrowOleException(long)	.text	004A32BE	00000032	R	.	L	.	.	.
sub_4A32F0	.text	004A32F0	0000003C	R	.	L	.	.	.
sub_4A332C	.text	004A332C	00000003	R
sub_4A332F	.text	004A332F	0000001C	R
sub_4A334B	.text	004A334B	00000036	R

```
100.00% (-316,-69,(663,534)) 000A9FEA:004A9FEA: WinMain(x,x,x,x)+8  
Executing function 'main'...  
Compiling file 'C:\Program Files\IDA Demo 5.0\idc\onload.idc'...  
Executing function 'OnLoad'...  
IDA is analysing the input file...  
You may start to explore the input file right now.  
Using FLIRT signature: Microsoft VisualC 2-8/net runtime  
Using FLIRT signature: MFC 3.1/4.0/4.2/8.0 32bit  
Reporting type information...
```

C++



The screenshot shows a 'Functions window' with a table of function symbols. The table has four columns: 'Function name', 'Segment', 'Start', and 'Length'. The functions listed include STL utility functions like do_scan_not, do_toupper, do_tolower, and collate_byname, as well as subroutines and filebuf destructors.

Function name	Segment	Start	Length
[_STL::ctype_byname<ushort>::do_scan_not(_STL::ctype_base::mask,us...	.text	004397B0	000000E4
[_STL::ctype_byname<ushort>::do_toupper(ushort)	.text	004398A0	00000014
[_STL::ctype_byname<ushort>::do_toupper(ushort*,ushort const*)	.text	004398C0	00000033
[_STL::ctype_byname<ushort>::do_tolower(ushort)	.text	00439900	00000014
[_STL::ctype_byname<ushort>::do_tolower(ushort*,ushort const*)	.text	00439920	00000033
[_STL::collate_byname<char>::collate_byname<char>(char const*,uint)	.text	00439960	00000075
sub_4399E0	.text	004399E0	00000054
[_STL::collate_byname<char>::~collate_byname<char>(void)	.text	00439A40	00000056
[_STL::collate_byname<char>::do_compare(char const*,char const*,char...	.text	00439AA0	00000027
[_STL::collate_byname<char>::do_transform(char const*,char const*)	.text	00439AD0	0000017C
[_STL::collate_byname<ushort>::collate_byname<ushort>(char const*,uint)	.text	00439C50	00000075
sub_439CD0	.text	00439CD0	00000054
[_STL::collate_byname<ushort>::~collate_byname<ushort>(void)	.text	00439D30	00000056
[_STL::collate_byname<ushort>::do_compare(ushort const*,ushort const*...	.text	00439D90	0000002B
[_STL::collate_byname<ushort>::do_transform(ushort const*,ushort const*)	.text	00439DC0	0000017C
[_STL::codecvt_byname<char,char,int>::codecvt_byname<char,char,int>(c...	.text	00439F40	0000001E
sub_439F60	.text	00439F60	00000054
[_STL::codecvt_byname<char,char,int>::~codecvt_byname<char,char,int>...	.text	00439FC0	0000000B
sub_439FD0	.text	00439FD0	00000075
filebuf::~scalar deleting destructor'(uint)	.text	0043A050	0000001E
filebuf::~filebuf(void)	.text	0043A070	00000056
sub_43A0D0	.text	0043A0D0	000000A1
sub_43A180	.text	0043A180	00000086

Line 1840 of 3304

C#/VB .NET

The screenshot displays the IDA Pro interface for a .NET assembly. The main window shows assembly code for a method named `hidebysig void FormMain_Load`. The code includes local variable declarations and calls to `ServerWatchDragon.FormMain::ClearDatas()` and `ServerWatchDragon.CEnvironment::FlagTrialEnd`.

The **Functions window** on the right lists various menu click handlers, such as `!menuNetworkMEMORY_Click`, `!menuNetworkPOP3_Click`, and `!menuServerAdd_Click`, with their respective segment, start, and length addresses.

The **Names window** on the far right shows a list of names, including `ServerWatchDragon.FormMailList`, `FormMail_Load`, `DisplayMailData`, and `FormMail_Load`.

The status bar at the bottom indicates the current address is `001E388` and the disk usage is `274MB`.

判断方法

- 各言語の特徴が出る
- Stringで入っているのでそこを見る
- バイナリエディッタでプログラムの最後のXMLを見る
- PEid(後述)を使う
 - 一番楽に結果が出ます

通信箇所を探す

- 目的は通信なので send() recv() を探してそこからたどるのみ。

Send 発見

The screenshot shows the IDA Pro interface with the following assembly code in the main window:

```
.text:005F311F      jz     short loc_5F315C
.text:005F3121      cmp    eax, 2
.text:005F3124      jz     short loc_5F315C
.text:005F3126      cmp    eax, 3
.text:005F3129      jz     short loc_5F315C
.text:005F312B      cmp    dword ptr [ecx+10h], 0
.text:005F312F      jnz   short loc_5F3152
.text:005F3131      mov    eax, [esp+flags]
.text:005F3135      mov    edx, [esp+len]
.text:005F3139      mov    ecx, [ecx+4]
.text:005F313C      push  eax                ; flags
.text:005F313D      mov    eax, [esp+4+buf]
.text:005F3141      push  edx                ; len
.text:005F3142      push  eax                ; buf
.text:005F3143      call  sub_4CB440
.text:005F3148      push  eax                ; s
.text:005F3149      call  ds:send
.text:005F314F      retn  0Ch
.text:005F3152      ; ~~~~~~
.text:005F3152      loc_5F3152:
.text:005F3152      mov    ecx, [ecx+10h]    ; CODE XREF: sub_5F30F0+3F↑j
.text:005F3155      mov    edx, [ecx]
```

The instruction `call ds:send` at address `005F3149` is highlighted in yellow. The Names window on the right shows a list of symbols, including `send` at offset `004A34B4`.

The status bar at the bottom indicates: `001F2539 | 005F3139: sub_5F30F0+49`

The console window at the bottom displays the following text:

```
Executing function 'OnLoad'...
IDA is analysing the input file...
You may start to explore the input file right now.
Using FLIRT signature: Microsoft VisualC 2-8/net runtime
Using FLIRT signature: MFC 3.1/4.0/4.2/8.0 32bit
Name '$LN7' at 004A34B4 is deleted...
Propagating type information...
Function argument information is propagated
The initial autoanalysis has been finished.
Retrieving information from the database... ok
AU: idle | Down | Disk: 437K
```

Sendを呼んでいるところの パラメータ

The screenshot shows the IDA Pro interface with the following assembly code in the main window:

```
; int __stdcall sub_5F30F0(char *buf,int len,int flags)
sub_5F30F0 proc near

buf= dword ptr 4
len= dword ptr 8
flags= dword ptr 0Ch

mov     eax, [ecx+0Ch]
test   eax, eax
jz     short loc_5F3104

push   eax                ; iError
call   ds:WSASetLastError
or     eax, 0FFFFFFFh

loc_5F3104:
mov     eax, [ecx+8]
```

The 'Names window' on the right lists various symbols, including 'CFormView::Create(wchar_t const*,wchar_t)'. The 'Strings window' shows a list of memory addresses and their corresponding string values, such as 'HEADER...', '00000027', 'C', 'IThis p', etc.

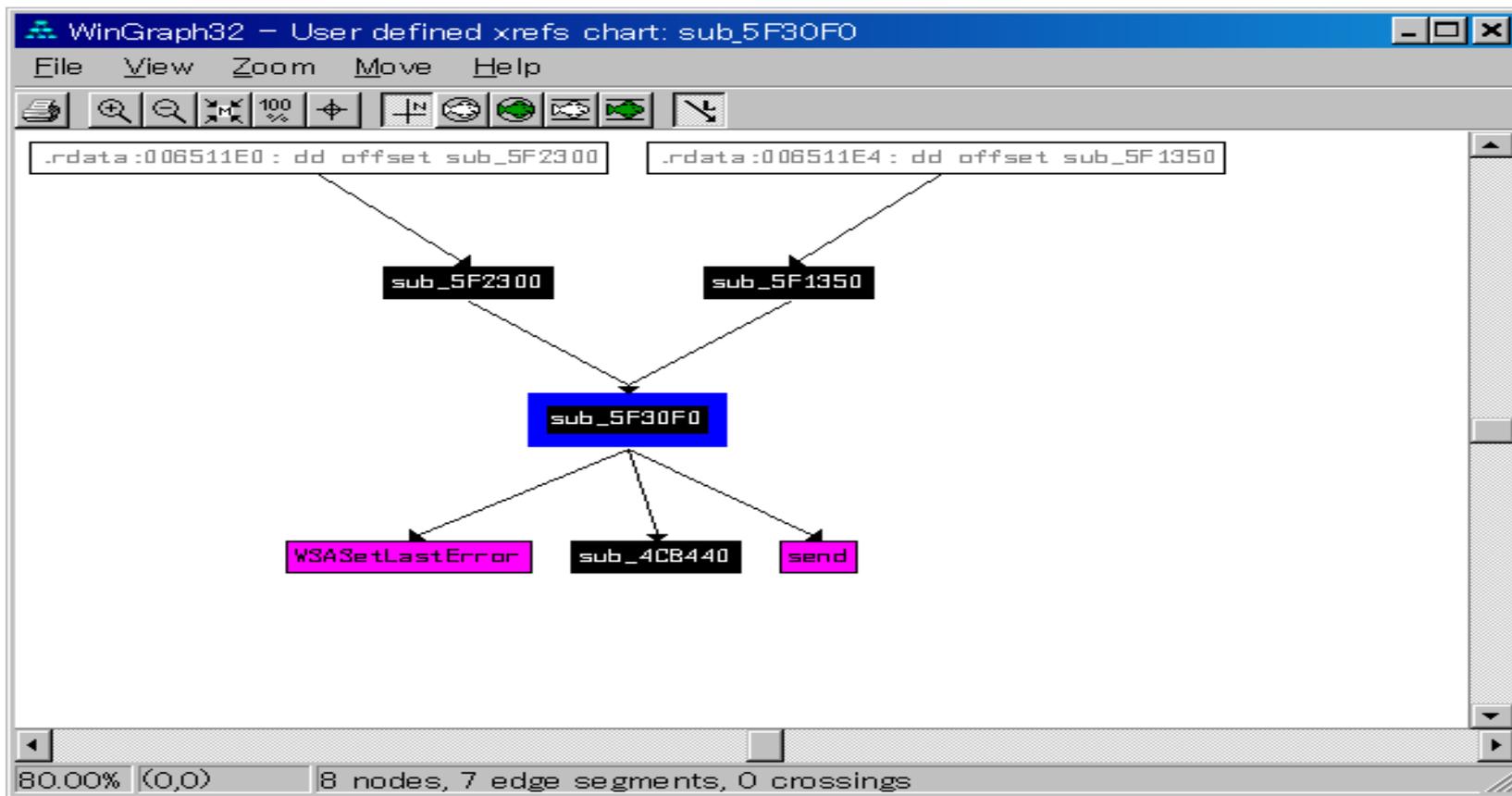
The status bar at the bottom indicates the current execution point: 'Executing function 'OnLoad'...' and 'Retrieving information from the database... ok'.

その関数の参照元

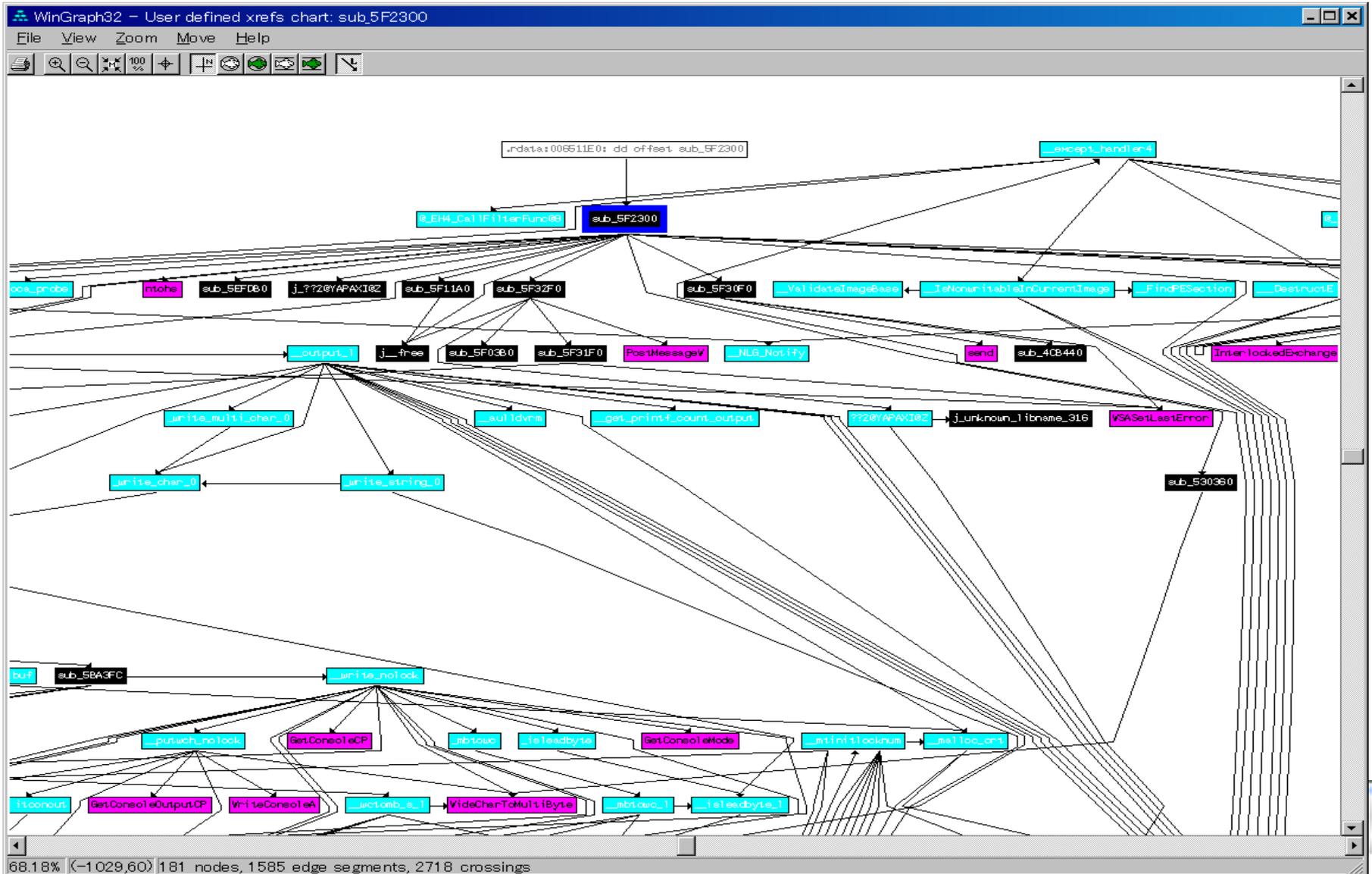
Dir...	T..	Address	Text
Up	j	sub_5F1270:loc_5F1287	jmp sub_5F30F0
Up	p	sub_5F1350+218	call sub_5F30F0
Up	p	sub_5F1350+3A0	call sub_5F30F0
Up	p	sub_5F1350+550	call sub_5F30F0
Up	p	sub_5F2300+130	call sub_5F30F0
Up	p	sub_5F2300+509	call sub_5F30F0
Up	p	sub_5F2300+6B1	call sub_5F30F0
Do...	j	sub_5F32E0	jmp sub_5F30F0

Line 1 of 8

その関数のチャート



1つまえの関数のチャート



動的解析(ollydbg)

- ディバック対策がされていると使えないことがある。
 - ディバック対策を回避するメモリパッチを作成
 - Olly上で当てて実行
 - プログラムが不安定になることも多い

呼び元がたどれない所は

- 動的解析をしてブレイクポイントをセットして待つ。
- 呼び元もしくは、サブルーチンの最後まで動かして戻るポイントが呼び元

実際にやってみます。

ネタ探し

- RootkitRevealer
- <http://www.microsoft.com/technet/sysinternals/utilities/RootkitRevealer.msp>
 - 自分のPCで発見されても困ります。

Packer パッカー

- UPX日本語フロントエンド
<http://www.vector.co.jp/soft/win95/util/se139612.html>
- UPX <http://upx.sourceforge.net/>
- ASPack
- <http://www.aspack.com/index.html>
- tElock
- <http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/TeLock.shtml>

UPX

The screenshot displays the IDA Pro interface for the file `H:\upx\p2ptest.exe`. The main window shows assembly code with a control flow graph overlaid. The assembly code includes instructions such as `mov ecx, edi`, `push edi`, `dec eax`, `repne scasb`, `push ebp`, `call dword ptr [esi+0E3968h]`, `or eax, eax`, and `jz short loc_4E2A4C`. The control flow graph shows a loop structure with branches to `loc_4E2A4C` and `loc_4E2A52`. A 'Graph overview' window provides a smaller view of the entire graph. The status bar at the bottom indicates the current function is `0005FE4C:004E2A4C: start:loc_4E2A4C`.

```
Executing function 'OnLoad'
IDA is analysing the input file...
You may start to explore the input file right now.
Propagating type information...
Function argument information is propagated
The initial autoanalysis has been finished.
UPX!0048300C: Can't find name (hint: use manual arg)
UPX!0048300C: Can't find name (hint: use manual arg)
Command "char|xrefsUser" failed
Command "char|xrefsFrom" failed
```

UPXのプログラム構造

UPXうさみみ

PID:00000B5C

Top - +

カテゴリ	Section Name	VirtualSize	VirtualAddress	SizeOfRawData	PointerToRawData	f
PEヘッダ	UPX0	00082000	00001000	00000000	00000400	(
セクション	UPX1	00060000	00083000	0005FC00	00000400	(
インポート関数	.rsrc	00002000	000E3000	00001A00	00060000	(
エクスポート関数						
実行ID						
実行状況						
スレッド						
レジスタ						
スタックトレース						
ハンドル						
ウィンドウ						
メモリマップ						
メモリ使用状況						
モジュール						
デバイスドライバ						
バス						
環境データ						
アクセストークン						
特権						
サービス						
ネットワーク						

自動更新 秒毎

列幅を最大字数で調整

Aspack

PID:00000C18

Top - +

カテゴリ	Section Name	VirtualSize	VirtualAddress	SizeOfRawData	PointerToRawData	PointerToRe
PEヘッダ	.text	00003000	00001000	00001400	00000600	00000000
セクション	.rdata	00002000	00004000	00005C200	00001A00	00000000
インポート関数	.data	00009000	00006000	00000200	0005DC00	00000000
エクスポート関数	.rsrc	00002000	000DF000	00000200	0005DE00	00000000
実行ID	.aspack	00003000	000E1000	00002A00	0005E000	00000000
実行状況	.adata	00001000	000E4000	00000000	00060A00	00000000
スレッド						
レジスタ						
スタックトレース						
ハンドル						
ウィンドウ						
メモリマップ						
メモリ使用状況						
モジュール						
デバイスドライバ						
バス						
環境データ						
アクセストークン						
特権						
サービス						
ネットワーク						

自動更新 3 秒毎
 列幅を最大字数で調整

ASprotect

PID:000007F8

Top - +

カテゴリ	Section Name	VirtualSize	VirtualAddress	SizeOfRawData	PointerToRawData	PointerToRe
PEヘッダ		00003000	00001000	00001400	00001000	00000000
セクション		00002000	00004000	0005C800	00002400	00000000
インポート関数		00009000	00006000	00000200	0005EA00	00000000
エクスポート関数	.rsrc	00002000	000DF000	00002000	0005EC00	00000000
実行ID		00001000	000E1000	00000200	00060C00	00000000
実行状況	.data	00018000	000E2000	00017C00	00060E00	00000000
スレッド	.adata	00001000	000FA000	00000000	00078A00	00000000
レジスタ						
スタックトレース						
ハンドル						
ウィンドウ						
メモリマップ						
メモリ使用状況						
モジュール						
デバイスドライバ						
バス						
環境データ						
アクセストークン						
特権						
サービス						
ネットワーク						

自動更新 秒毎
 列幅を最大字数で調整

PEiD

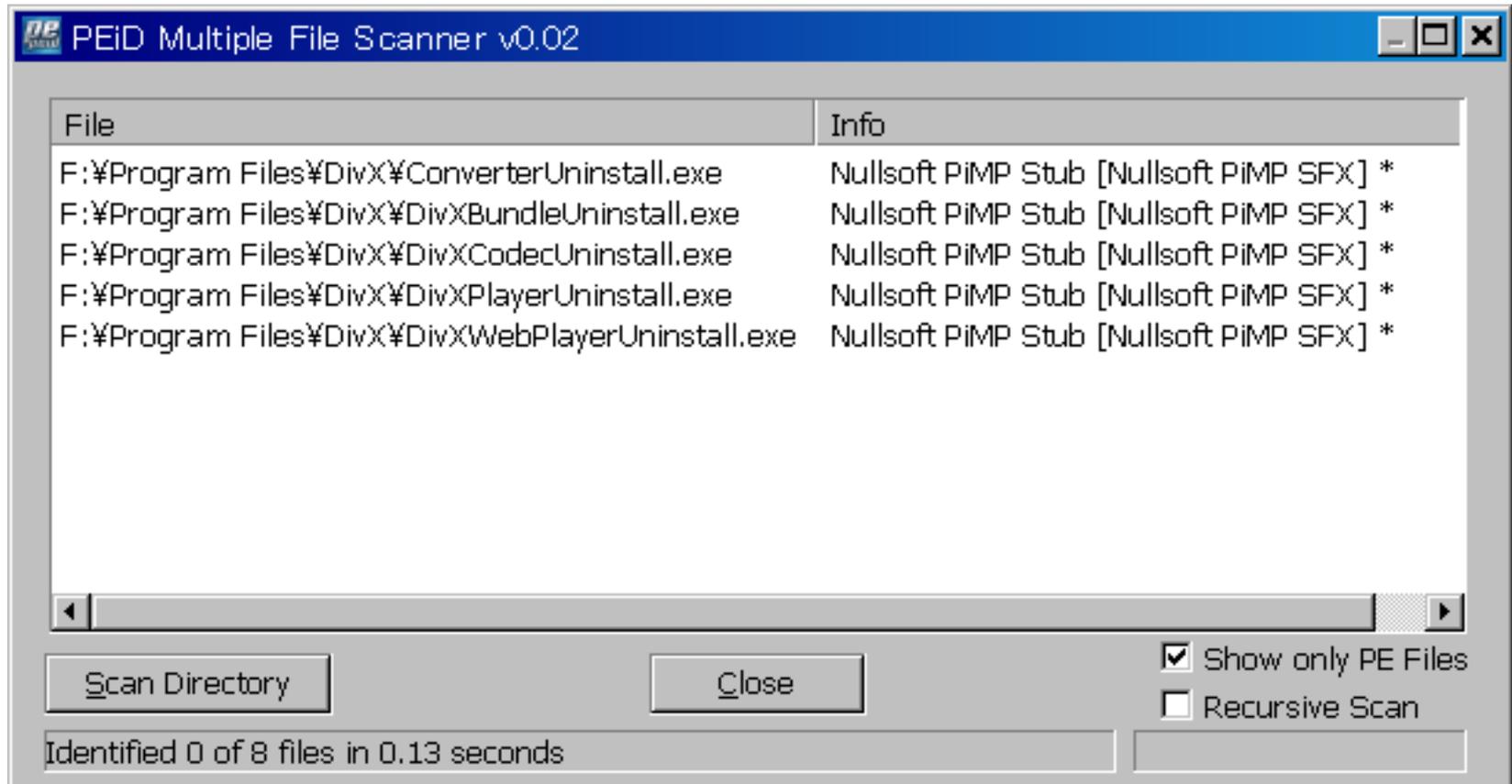
– パッカーのチェック

- <http://www.secretashell.com/codomain/peid/>

– 結果のサンプル用

- Lhaplus
- Winrar
- DivX
- Xvid

PEid結果



動的解析とメモリ情報の取得

- Ollyを使ってやってみましょう

暗号サブルーチンの発見(IDA)

- IDAのプラグインであたりをつけましょう。

暗号前のデータを手

- Ollyでやってみましょう。

まとめ

- バイナリプログラムでもかなり用意に通信内容を解析できます。
- パケットキャプチャでできるならそっちの方が楽にできる。
- 暗号化されていても通信しているプログラムは実態を持っている。

Net
Agent